CLAIMS

What is claimed is:

1.   A method comprising:
     defining a domain-specific language usable in a modeling environment and having a
     dynamic component and a static component, said dynamic component able to affect a
     behavior of said static component.

2.   The method of claim 1, wherein defining said domain-specific language comprises:
     defining said dynamic component and said static component in accordance with Unified
     Modeling Language constructs and semantics.

3.   The method of claim 1, wherein defining said domain-specific language comprises:
     defining a customized Unified Modeling Language meta-modeling profile which supports
     definitions of said dynamic component and said static component.

4.   The method of claim 1, wherein defining said domain-specific language comprises:
     defining said domain-specific language based on custom meta-modeling constructs, said
     constructs in accordance with a Unified Modeling Language meta-modeling profile and
     defining said dynamic component and said static component.

5.   The method of claim 1, wherein defining said domain-specific language comprises:
     importing a definition of an element of said domain-specific language from a previously-
     defined domain-specific language.

6.   The method of claim 1, wherein defining said domain-specific language comprises:
     validating said domain-specific language in accordance with a validation rule defined in a
     meta-modeling language.

7.   The method of claim 1, wherein defining said domain-specific language comprises:

generating an eXtensible Markup Language output representing at least one definition of said domain-specific language.

8. The method of claim 1, wherein defining said domain-specific language comprises:
    defining a custom action available for execution, on an element of an application model compliant with said domain-specific language, in response to an invocation request in accordance with said domain-specific language.

9. The method of claim 1, wherein defining said domain-specific language comprises:
    defining at least one language information item of said domain-specific language;
    defining at least one language term of said domain-specific language; and
    defining at least one data type of said domain-specific language.

10. The method of claim 9, wherein defining said domain-specific language further comprises:
    defining a relationship between said at least one language term and another language term of said domain-specific language.

11. The method of claim 9, wherein defining said domain-specific language further comprises:
    defining a constraint associated with one or more elements of said domain-specific language to be used during validation of said one or more elements of said domain-specific language.

12. The method of claim 9, wherein defining said domain-specific language further comprises:
    defining an aspect able to affect an element selected from a group consisting of: said at least one language term, a property of said at least one language term, and a relationship between said at least one language term and another language term.

13. The method of claim 1, further comprising:
    applying said domain-specific language to said model during execution of a modeling process of said model.

14. The method of claim 13, further comprising:

    creating one or more elements of a model in accordance with at least one language term defined in said domain-specific language.

15. The method of claim 14, wherein creating comprises:

    generating a recommended modeling route to be used during creation of said one or more elements of said model in accordance with a mentor modeling definition of said domain-specific language.

16. The method of claim 14, wherein creating comprises:

    executing a custom action defined in said domain-specific language on at least one of said one or more elements of said model.

17. The method of claim 14, wherein creating comprises:

    converting a domain-specific model artifact of said domain-specific language into an application artifact usable during execution of said modeling process.

18. The method of claim 17, further comprising:

    storing said domain-specific model artifact in a metadata database able to provide access to said domain-specific model artifact.

19. A system for accelerated modeling, the system comprising:

    a language builder module to define a domain-specific language usable in a modeling environment and having a dynamic component and a static component, said dynamic component able to affect a behavior of said static component.

20. The system of claim 19, wherein said language builder module is able to import a definition of an element of said domain-specific language from a previously-defined domain-specific language.

21. The system of claim 19, wherein said language builder module comprises a validator to validate said domain-specific language in accordance with a validation rule defined in a meta-modeling language.

22. The system of claim 19, wherein said language builder module comprises a generator to generate an eXtensible Markup Language output representing at least one definition of said domain-specific language.

23. The system of claim 19, wherein said language builder module comprises an action editor to define a custom action available for execution on a model in accordance with said domain-specific language in response to an invocation request in accordance with said domain-specific language.

24. The system of claim 19, wherein said language builder module is able to define at least one language information item of said domain-specific language, to define at least one language term of said domain-specific language, and to define at least one data type of said domain-specific language.

25. The system of claim 24, wherein said language builder module is able to define a relationship between said at least one language term and another language term of said domain-specific language.

26. The system of claim 19, wherein said language builder module comprises a constraint editor to define a constraint associated with said data type to be used during validation of one or more elements of said domain-specific language.

27. The system of claim 19, comprising a mentoring module to generate a recommended modeling route available during creation of one or more elements of a model in accordance with a mentor modeling definition of said domain-specific language.

28. The system of claim 19, comprising a generator able to create one or more elements of a model in accordance with a process defined in said domain-specific language.

29. The system of claim 28, comprising a language runtime module to apply said domain-specific language to said model during execution of a runtime process of said model.

30. The system of claim 29, wherein said language runtime module comprises a validator to validate said model based on a validation rule defined in said domain-specific language.

31. The system of claim 29 wherein said language runtime module comprises an action executor to execute a custom action defined in said domain-specific language on at least one of said one or more elements of said model.

32. The system of claim 29, wherein said language runtime module comprises a process mentor module to guide said runtime process in accordance with a process definition of said domain-specific language.

33. The system of claim 29, wherein said language runtime module comprises a generator to generate an eXtensible Markup Language output representing said model based on said domain-specific language.

34. The system of claim 29, comprising a converter to convert a domain-specific model artifact based on said domain-specific language into an application artifact usable during execution of said runtime process.

35. The system of claim 34, comprising a database to store said domain-specific model artifact and to provide access to said domain-specific model artifact during execution of said runtime process.

36. A machine-readable medium having stored thereon instructions that, when executed by a machine, result in:

37

defining a domain-specific language usable in a modeling environment and having a dynamic component and a static component, said dynamic component able to affect a behavior of said static component.

37.   The machine-readable medium of claim 36, wherein the instructions result in:
defining said dynamic component and said static component in accordance with Unified Modeling Language constructs and semantics.

38.   The machine-readable medium of claim 36, wherein the instructions result in:
defining a customized Unified Modeling Language meta-modeling profile which supports definitions of said dynamic component and said static component.

39.   The machine-readable medium of claim 36, wherein the instructions result in:
defining said domain-specific language based on custom meta-modeling constructs, said constructs in accordance with a Unified Modeling Language meta-modeling profile and defining said dynamic component and said static component.